Міністерство освіти та науки України Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет прикладної математики

Кафедра системного програмування i спеціалізованих комп'ютерних систем

Лабораторна робота №1



“ Бази даних і засоби управління”

Тема: “ Проектування бази даних та ознайомлення з базовими операціями СУБД PostgreSQL”

Виконав: Шевченко О. В.

Студент групи: КВ-93

Перевірив(ла):

Київ 2021

**Лабораторна робота №1**

Проектування бази даних та ознайомлення з базовими операціями СУБД PostgreSQL

Метою роботи є здобуття вмінь проектування бази даних та практичних навичок створення реляційних баз даних за допомогою PostgreSQL.

Завдання роботи полягає у наступному:

1. Розробити модель «сутність-зв’язок» предметної галузі, обраної студентом самостійно.
2. Перетворити розроблену модель у схему бази даних (таблиці) PostgreSQL.
3. Виконати нормалізацію схеми бази даних до третьої нормальної форми (3НФ).
4. Ознайомитись із інструментарієм PostgreSQL та pgAdmin 4 та внести декілька рядків даних у кожну з таблиць засобами pgAdmin 4.

**Варіант (назва теми): «Автосалон»**

Обрана предметна галузь передбачає отримання і обробку замовлень від покупців. Згідно цієї області для побудови бази даних було виділено наступні сутності:

Покупець(Customers), з атрибутами: код, ім’я замовника, прізвище, телефон. Призначена для того, щоб містити контакти замовника або покупця.

Автомобілі(Cars), з атрибутами: номер, назва бренду, назва моделі, рік виготовлення. Призначена для розподілу і зберігання автомобілів які йдуть на продаж.

Замовлення(Orders), з атрибутами: номер замовлення, код покупця, код автомобіля, дата. Призначена для фіксування факту оформлення замовлення.

Продавець(Stuff), з атрибутами: код, прізвище, телефон, дата початку роботи. Призначена для збереження даних тих магазинів, для яких здійснюється обробка замовлень.

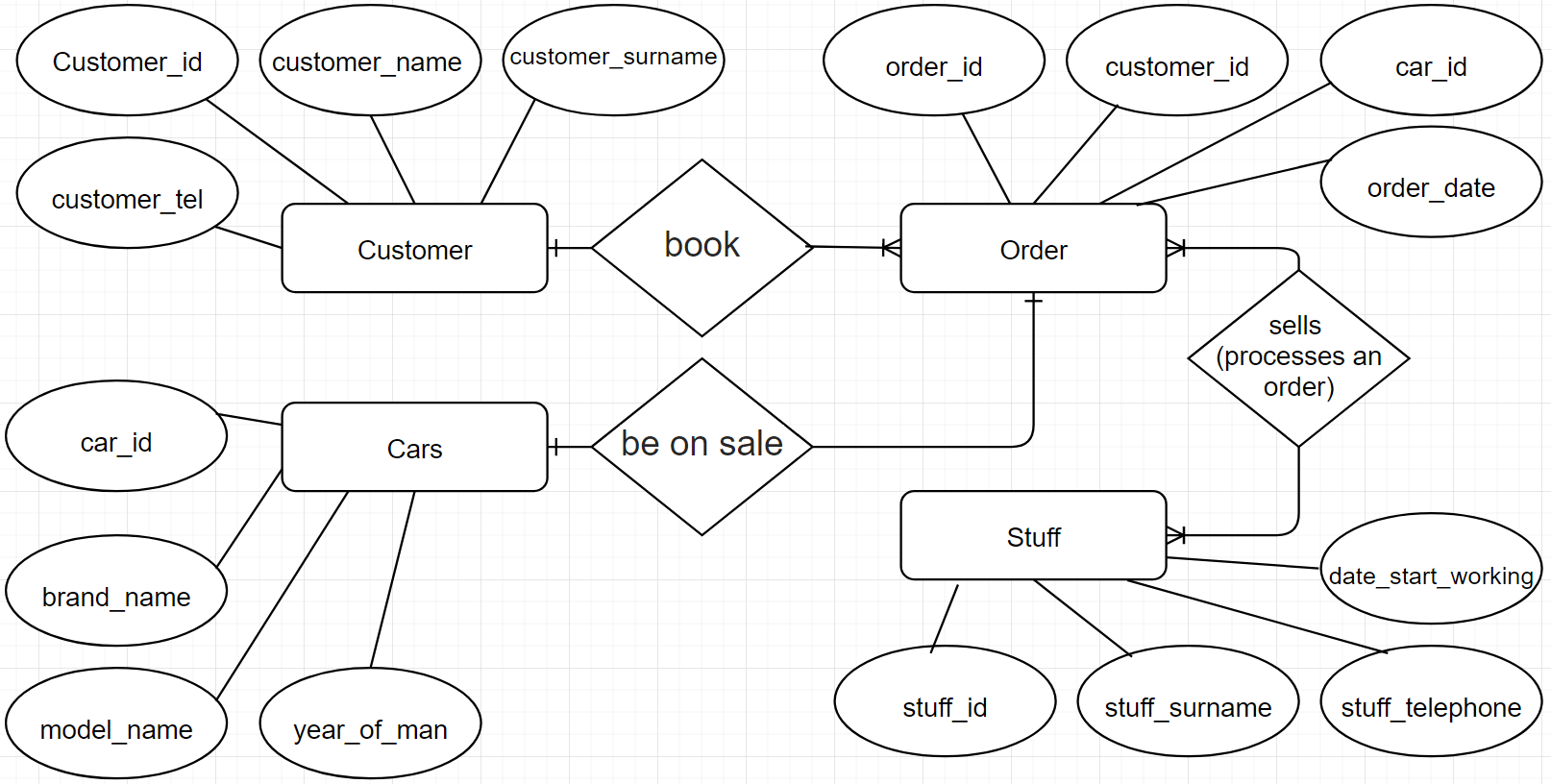


Рисунок 1 - ER-діаграма, побудована за нотацією Чена

Модель «сутність-зв’язок» нарисована за допомогою інструменту <https://www.draw.io/>

Опис зв'язків

Маємо зв’язок-сутність «Замовлення» між 3-ма сутностями: «Покупець», «Автомобіль» і «Продавець».

Сутність «Покупець» має зв’язок 1:N по відношенню до «Замовлення», тому що в одному замовленні може приймати участь тільки один покупець, але один покупець може виконувати різну кількість замовлень

Сутність «Автомобіль» має зв’язок 1:1 по відношенню до «Замовлення», тому що в одному замовленні може знаходитись тільки один автомобіль і на кожен автомобіль припадає тільки одне замовлення.

Сутність «Продавець» має зв’язок 1:N по відношенню до «Замовлення», тому що одне замовлення може виконувати тільки один продавець і при цьому продавці можуть фігурувати у декількох різних замовленнях.

**Опис структури БД «Автосалон»**

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип(Розмір) |
| Customers (вміщує інформацію про покупця) | customer\_id – унікальний ID покупця  customer\_name – ім’я покупця  customer\_surname – фамілія покупця  customer\_telephone – телефон покупця | Числовий  Текстовий  Текстовий  Текстовий(14) |
| Cars (вміщує інформацію про автомобілі) | car\_id - унікальний ID автомобіля  brand\_name – ім’я бренду автомобіля  model\_name – ім’я модельного ряду автомобіля  year\_of\_manufacture – рік виготовлення автомобіля | Числовий  Текстовий  Текстовий  Числовий |
| Orders (вміщує інформацію про зроблене замовлення) | order\_id - унікальний ID замовлення  customer\_id – ID покупця  car\_id - ID автомобіля  order\_date – дата замовлення | Числовий  Числовий  Числовий  Дата |
| Stuff (вміщує інформацію про продавців) | stuff\_id - унікальний ID персоналу  stuff\_surname - ім’я продавця  stuff\_telephone – телефон продавця  date\_start\_working – коли почав працювати продавець | Числовий  Текстовий  Текстовий(14)  Дата |

**Перетворена ER-діаграма у ТАБЛИЦІ БД**

Перетворені та зведені до третьої нормальної форми (ЗНФ) таблиці БД

Cутність Customer було перетворено у таблицю Customer

Cутність Car було перетворено у таблицю Car

Cутність-зв’язок Order було перетворено у таблицю Order, таким чином, щоб атрибути car\_id, customer\_id, stuff\_id мали вторинні ключі у таблиці Order.

Cутність Stuff було перетворено у таблицю Stuff

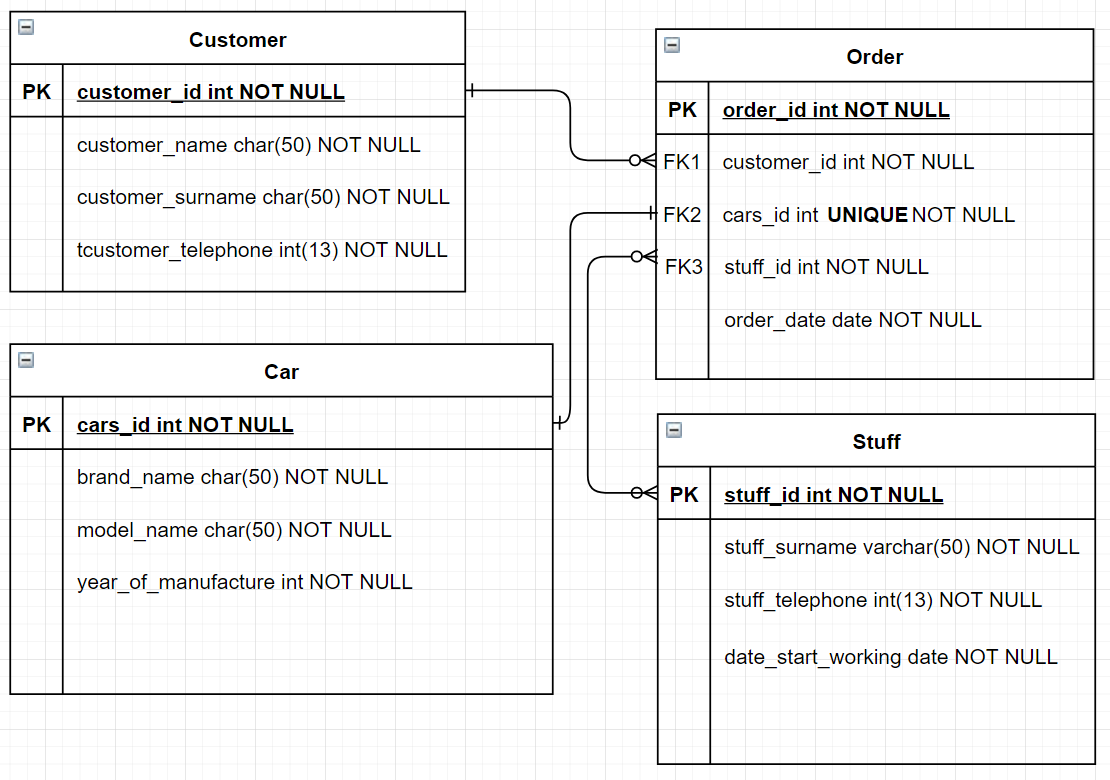


Рисунок 2 - ER-діаграма, перетворена у таблиці БД

**Нормалізація схеми бази даних до третьої нормальної форми**

Схема відповідає 1НФ, тому що:

1) Кожна таблиця має основний ключ (Primary key, PK) та мінімальний набір атрибутів, які ідентифікують запис;

2) Кожен атрибут має лише одне значення, а не множину значень.

Схема відповідає 2НФ, тому що:

1) Вона відповідає вимогам 1НФ;

2) Кожен не ключовий атрибут функціонально залежить від основного атрибуту.

Схема відповідає 3НФ, а саме:

1) Вона відповідає вимогам 2НФ;

2) Дані в таблиці залежать винятково від основного ключа

**Функціональні залежності для кожної таблиці**

Customer**:**

customer\_id - customer\_name, customer\_surname, customer\_telephone

customer\_id - customer\_name (ім’я покупця, яке пов’язане з його ідентифікатором)

customer\_id - customer\_surname (прізвище покупця, яке пов’язане з його ідентифікатором)

customer\_id - customer\_telephone (телефон покупця, який пов’язаний з його ідентифікатором)

customer\_id - customer\_name - customer\_surname - customer\_telephone (транзитивне відношення, але воно включає в себе ключовий атрибут)

customer\_id - customer\_name - customer\_telephone - customer\_surname (транзитивне відношення, але воно включає в себе ключовий атрибут)

customer\_id - customer\_surname - customer\_name - customer\_telephone (транзитивне відношення, але воно включає в себе ключовий атрибут)

customer\_id - customer\_surname - customer\_telephone - customer\_name (транзитивне відношення, але воно включає в себе ключовий атрибут)

customer\_id - customer\_telephone - customer\_surname - customer\_name (транзитивне відношення, але воно включає в себе ключовий атрибут)

customer\_id - customer\_telephone - customer\_name - customer\_surname (транзитивне відношення, але воно включає в себе ключовий атрибут)

Car:

car\_id - brand\_name, model\_name, year\_of\_manufacture

car\_id - brand\_name (ім’я марки автомобіля, яка пов’язана з його ідентифікатором)

car\_id - model\_name (ім’я моделі автомобіля, яка пов’язана з його ідентифікатором)

car\_id - year\_of\_manufacture (рік виготовлення автомобіля, який пов’язаний з його ідентифікатором)

car\_id - brand\_name - model\_name - year\_of\_manufacture (транзитивне відношення, але воно включає в себе ключовий атрибут)

car\_id - brand\_name - year\_of\_manufacture - model\_name (транзитивне відношення, але воно включає в себе ключовий атрибут)

car\_id - model\_name - brand\_name - year\_of\_manufacture (транзитивне відношення, але воно включає в себе ключовий атрибут)

car\_id - model\_name - year\_of\_manufacture - brand\_name (транзитивне відношення, але воно включає в себе ключовий атрибут)

car\_id - year\_of\_manufacture - model\_name - brand\_name (транзитивне відношення, але воно включає в себе ключовий атрибут)

car\_id - year\_of\_manufacture - brand\_name - model\_name (транзитивне відношення, але воно включає в себе ключовий атрибут)

Stuff:

stuff\_id - stuff\_surname, stuff\_telephone, date\_start\_working

stuff\_id - stuff\_surname (прізвище продавця, яке пов’язане з його ідентифікатором)

stuff\_id - stuff\_telephone (телефон продавця, який також пов’язується з його ідентифікатором)

stuff\_id - date\_start\_working (дата початку роботи продавця, яка пов’язана з його ідентифікатором)

stuff\_id - stuff\_surname - stuff\_telephone - date\_start\_working (транзитивне відношення, але воно включає в себе ключовий атрибут)

stuff\_id - stuff\_surname - date\_start\_working - stuff\_telephone (транзитивне відношення, але воно включає в себе ключовий атрибут)

stuff\_id - stuff\_telephone - stuff\_surname - date\_start\_working (транзитивне відношення, але воно включає в себе ключовий атрибут)

stuff\_id - stuff\_telephone - date\_start\_working - stuff\_surname (транзитивне відношення, але воно включає в себе ключовий атрибут)

stuff\_id - date\_start\_working - stuff\_telephone - stuff\_surname (транзитивне відношення, але воно включає в себе ключовий атрибут)

stuff\_id - date\_start\_working - stuff\_surname - stuff\_telephone (транзитивне відношення, але воно включає в себе ключовий атрибут)

Order:

order\_id - order\_date

order\_id - order\_date (дата замовлення, яке пов’язане з його ідентифікатором)

order\_id - order\_date (транзитивне відношення, але воно включає в себе ключовий атрибут)

**Перетворена ER-діаграма у ТАБЛИЦІ БД**

|  |  |  |  |
| --- | --- | --- | --- |
| Customer | | | |
| customer\_id | customer\_name | customer\_surname | customer\_telephone |
| 1 | Yevheniia | Baranova | 044-300-20-32 |
| 2 | Oleksandr | Bubriak | 044-363-41-73 |
| 3 | Volodymyr | Vitkovskyi | 099-180-76-92 |
| 4 | Yusupov | Oleksandr | 068-897-11-13 |
| 5 | Skyba | Yaroslav | 099-969-12-67 |

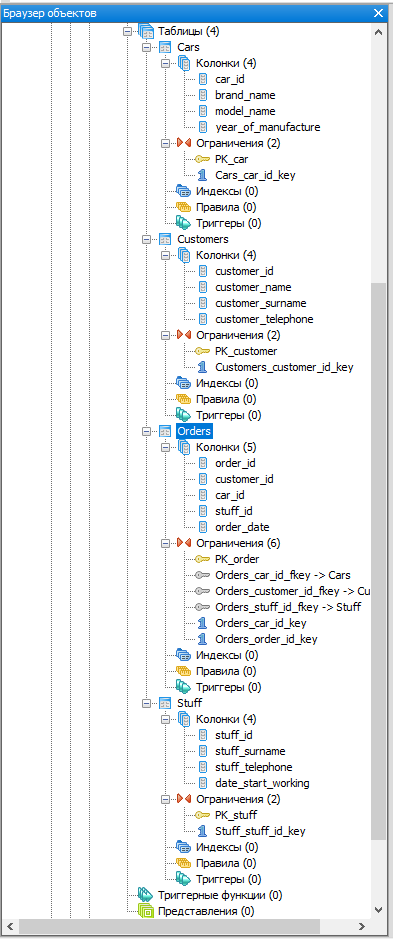
|  |  |  |  |
| --- | --- | --- | --- |
| Car | | | |
| car\_id | brand\_name | model\_name | year\_of\_manufacture |
| 1 | KIA | Rio | 2015 |
| 2 | Honda | Solaris | 2019 |
| 3 | BMW | X5 | 2010 |
| 4 | BMW | ML | 2012 |
| 5 | Mazda | 6 | 2007 |
| 6 | Volkswagen | Passat | 2014 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Order | | | | |
| order\_id | customer\_id | car\_id | stuff\_id | order\_date |
| 1 | 2 | 1 | 3 | 17-12-2019 |
| 2 | 3 | 2 | 1 | 13-11-2019 |
| 3 | 5 | 3 | 2 | 17-03-2020 |
| 4 | 2 | 4 | 3 | 26-03-2020 |
| 5 | 4 | 5 | 1 | 19-05-2020 |
| 6 | 1 | 6 | 1 | 16-03-2021 |

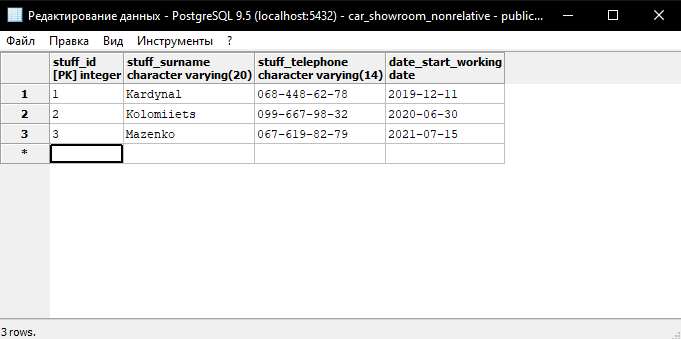
|  |  |  |  |
| --- | --- | --- | --- |
| Stuff | | | |
| stuff\_id | stuff\_surname | stuff\_telephone | date\_start\_working |
| 1 | Kardynal | 068-448-62-78 | 11-10-2019 |
| 2 | Kolomiiets | 099-667-98-32 | 30-06-2020 |
| 3 | Mazenko | 067-619-82-79 | 15-07-2021 |

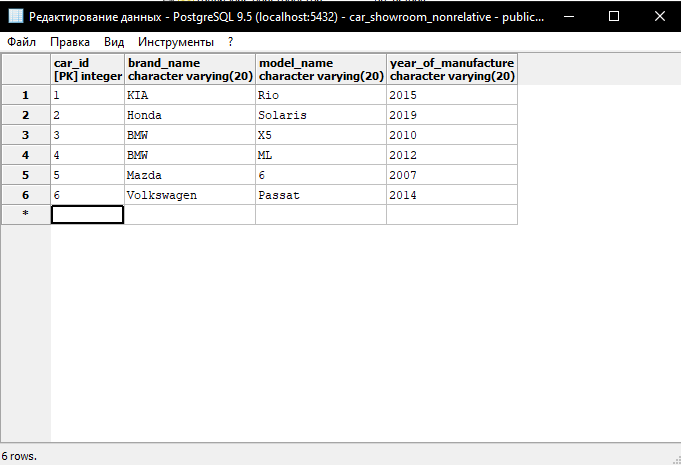
Синім кольором у таблиці виділені стовпці з первинними ключами, а червоним – вторинними ключами. Стрілки вказують на залежність між первинними та вторинними ключами.

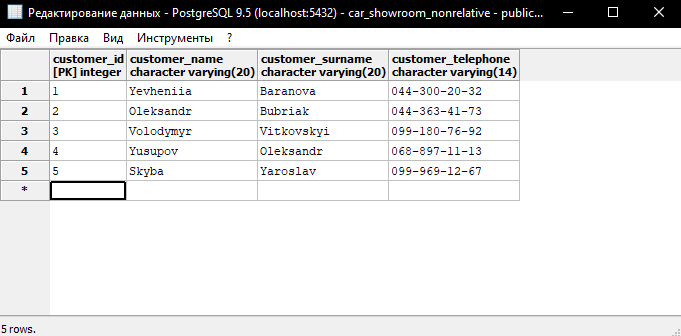
**Копії екрану з pgAdmin3, що відображають назви, типи та обмеження на стовпці**

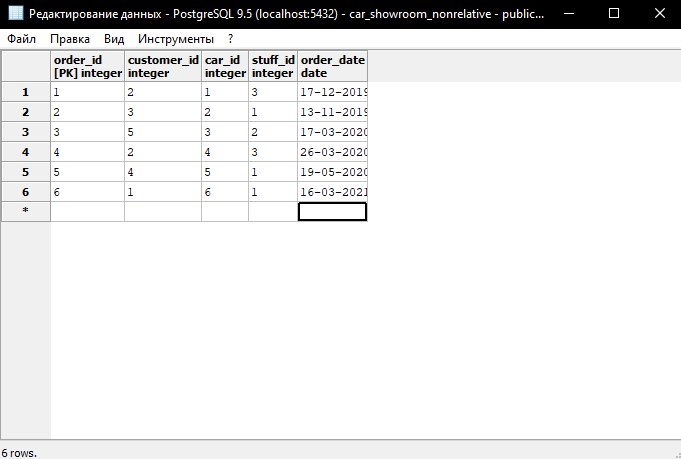


**Копії екрану з pgAdmin3, що відображають вміст таблиць бази даних у PostgreSQL**

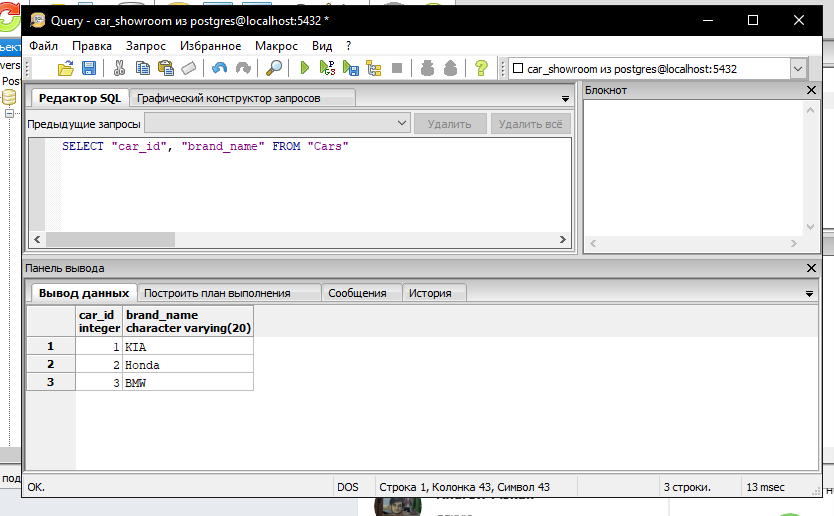








**Приклад SQL запиту**



**SQL script of DB**

SET statement\_timeout = 0;

SET lock\_timeout = 0;

SET client\_encoding = 'UTF8';

SET standard\_conforming\_strings = on;

SELECT pg\_catalog.set\_config('search\_path', '', false);

SET check\_function\_bodies = false;

SET xmloption = content;

SET client\_min\_messages = warning;

SET row\_security = off;

CREATE EXTENSION IF NOT EXISTS plpgsql WITH SCHEMA pg\_catalog;

COMMENT ON EXTENSION plpgsql IS 'PL/pgSQL procedural language';

SET default\_tablespace = '';

SET default\_with\_oids = false;

CREATE TABLE public."Cars" (

car\_id integer NOT NULL,

brand\_name character varying(20) NOT NULL,

model\_name character varying(20) NOT NULL,

year\_of\_manufacture character varying(20) NOT NULL

);

ALTER TABLE public."Cars" OWNER TO postgres;

CREATE SEQUENCE public."Cars\_car\_id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public."Cars\_car\_id\_seq" OWNER TO postgres;

ALTER SEQUENCE public."Cars\_car\_id\_seq" OWNED BY public."Cars".car\_id;

CREATE TABLE public."Customers" (

customer\_id integer NOT NULL,

customer\_name character varying(20) NOT NULL,

customer\_surname character varying(20) NOT NULL,

customer\_telephone character varying(14) NOT NULL

);

ALTER TABLE public."Customers" OWNER TO postgres;

CREATE SEQUENCE public."Customers\_customer\_id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public."Customers\_customer\_id\_seq" OWNER TO postgres;

ALTER SEQUENCE public."Customers\_customer\_id\_seq" OWNED BY public."Customers".customer\_id;

CREATE TABLE public."Orders" (

order\_id integer NOT NULL,

customer\_id integer NOT NULL,

car\_id integer NOT NULL,

stuff\_id integer NOT NULL,

order\_date date

);

ALTER TABLE public."Orders" OWNER TO postgres;

CREATE SEQUENCE public."Orders\_car\_id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public."Orders\_car\_id\_seq" OWNER TO postgres;

ALTER SEQUENCE public."Orders\_car\_id\_seq" OWNED BY public."Orders".car\_id;

CREATE SEQUENCE public."Orders\_customer\_id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public."Orders\_customer\_id\_seq" OWNER TO postgres;

ALTER SEQUENCE public."Orders\_customer\_id\_seq" OWNED BY public."Orders".customer\_id;

CREATE SEQUENCE public."Orders\_order\_id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public."Orders\_order\_id\_seq" OWNER TO postgres;

ALTER SEQUENCE public."Orders\_order\_id\_seq" OWNED BY public."Orders".order\_id;

CREATE SEQUENCE public."Orders\_stuff\_id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public."Orders\_stuff\_id\_seq" OWNER TO postgres;

ALTER SEQUENCE public."Orders\_stuff\_id\_seq" OWNED BY public."Orders".stuff\_id;

CREATE TABLE public."Stuff" (

stuff\_id integer NOT NULL,

stuff\_surname character varying(20) NOT NULL,

stuff\_telephone character varying(14) NOT NULL,

date\_start\_working date NOT NULL

);

ALTER TABLE public."Stuff" OWNER TO postgres;

CREATE SEQUENCE public."Stuff\_stuff\_id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public."Stuff\_stuff\_id\_seq" OWNER TO postgres;

ALTER SEQUENCE public."Stuff\_stuff\_id\_seq" OWNED BY public."Stuff".stuff\_id;

ALTER TABLE ONLY public."Cars" ALTER COLUMN car\_id SET DEFAULT nextval('public."Cars\_car\_id\_seq"'::regclass);

ALTER TABLE ONLY public."Customers" ALTER COLUMN customer\_id SET DEFAULT nextval('public."Customers\_customer\_id\_seq"'::regclass);

ALTER TABLE ONLY public."Orders" ALTER COLUMN order\_id SET DEFAULT nextval('public."Orders\_order\_id\_seq"'::regclass);

ALTER TABLE ONLY public."Orders" ALTER COLUMN customer\_id SET DEFAULT nextval('public."Orders\_customer\_id\_seq"'::regclass);

ALTER TABLE ONLY public."Orders" ALTER COLUMN car\_id SET DEFAULT nextval('public."Orders\_car\_id\_seq"'::regclass);

ALTER TABLE ONLY public."Orders" ALTER COLUMN stuff\_id SET DEFAULT nextval('public."Orders\_stuff\_id\_seq"'::regclass);

ALTER TABLE ONLY public."Stuff" ALTER COLUMN stuff\_id SET DEFAULT nextval('public."Stuff\_stuff\_id\_seq"'::regclass);

COPY public."Cars" (car\_id, brand\_name, model\_name, year\_of\_manufacture) FROM stdin;

1 KIA Rio 2015

2 Honda Solaris 2019

3 BMW X5 2010

4 BMW ML 2012

5 Mazda 6 2007

6 Volkswagen Passat 2014

\.

SELECT pg\_catalog.setval('public."Cars\_car\_id\_seq"', 1, false);

COPY public."Customers" (customer\_id, customer\_name, customer\_surname, customer\_telephone) FROM stdin;

1 Yevheniia Baranova 044-300-20-32

2 Oleksandr Bubriak 044-363-41-73

3 Volodymyr Vitkovskyi 099-180-76-92

4 Yusupov Oleksandr 068-897-11-13

5 Skyba Yaroslav 099-969-12-67

\.

SELECT pg\_catalog.setval('public."Customers\_customer\_id\_seq"', 1, false);

COPY public."Orders" (order\_id, customer\_id, car\_id, stuff\_id, order\_date) FROM stdin;

1 2 1 3 2019-12-17

2 3 2 1 2019-11-13

3 5 3 2 2020-03-17

4 2 4 3 2020-03-26

5 4 5 1 2020-05-19

6 1 6 1 2021-03-16

\.

SELECT pg\_catalog.setval('public."Orders\_car\_id\_seq"', 9, true);

SELECT pg\_catalog.setval('public."Orders\_customer\_id\_seq"', 9, true);

SELECT pg\_catalog.setval('public."Orders\_order\_id\_seq"', 1, true);

SELECT pg\_catalog.setval('public."Orders\_stuff\_id\_seq"', 16, true);

COPY public."Stuff" (stuff\_id, stuff\_surname, stuff\_telephone, date\_start\_working) FROM stdin;

1 Kardynal 068-448-62-78 2019-12-11

2 Kolomiiets 099-667-98-32 2020-06-30

3 Mazenko 067-619-82-79 2021-07-15

\.

SELECT pg\_catalog.setval('public."Stuff\_stuff\_id\_seq"', 1, false);

ALTER TABLE ONLY public."Cars"

ADD CONSTRAINT "Cars\_car\_id\_key" UNIQUE (car\_id);

ALTER TABLE ONLY public."Customers"

ADD CONSTRAINT "Customers\_customer\_id\_key" UNIQUE (customer\_id);

ALTER TABLE ONLY public."Orders"

ADD CONSTRAINT "Orders\_car\_id\_key" UNIQUE (car\_id);

ALTER TABLE ONLY public."Orders"

ADD CONSTRAINT "Orders\_order\_id\_key" UNIQUE (order\_id);

ALTER TABLE ONLY public."Cars"

ADD CONSTRAINT "PK\_car" PRIMARY KEY (car\_id);

ALTER TABLE ONLY public."Customers"

ADD CONSTRAINT "PK\_customer" PRIMARY KEY (customer\_id);

ALTER TABLE ONLY public."Orders"

ADD CONSTRAINT "PK\_order" PRIMARY KEY (order\_id);

ALTER TABLE ONLY public."Stuff"

ADD CONSTRAINT "PK\_stuff" PRIMARY KEY (stuff\_id);

ALTER TABLE ONLY public."Stuff"

ADD CONSTRAINT "Stuff\_stuff\_id\_key" UNIQUE (stuff\_id);

ALTER TABLE ONLY public."Orders"

ADD CONSTRAINT "Orders\_car\_id\_fkey" FOREIGN KEY (car\_id) REFERENCES public."Cars"(car\_id);

ALTER TABLE ONLY public."Orders"

ADD CONSTRAINT "Orders\_customer\_id\_fkey" FOREIGN KEY (customer\_id) REFERENCES public."Customers"(customer\_id);

ALTER TABLE ONLY public."Orders"

ADD CONSTRAINT "Orders\_stuff\_id\_fkey" FOREIGN KEY (stuff\_id) REFERENCES public."Stuff"(stuff\_id);

REVOKE ALL ON SCHEMA public FROM PUBLIC;

REVOKE ALL ON SCHEMA public FROM postgres;

GRANT ALL ON SCHEMA public TO postgres;

GRANT ALL ON SCHEMA public TO PUBLIC;